

Técnicas Actuales de Programación

2018/2019 Examen Final

1- Raíz cuadrada (1/4)

Los Babilonios calculaban la raíz cuadrada mediante una aproximación:

$$\sqrt{X} \simeq \frac{X + N^2}{2N}$$

En el manuscrito de Bakhshali (siglo III o IV) se encuentra otro método para este mismo cálculo:

$$\sqrt{X} \simeq \frac{N^4 + 6N^2X + X^2}{4N^3 + 4NX}$$

Ambas comienzan por encontrar de un modo “manual” el entero (N) que más se aproxima a X al elevarlo al cuadrado para aplicarlo posteriormente en la fórmula.

Escriba una clase que proporcione dos métodos para calcular la raíz cuadrada mediante las dos aproximaciones, y en el método `main()` compruebe su funcionamiento para los valores de X entre 16 y 25 imprimiendo el error que cometen frente al valor que se obtiene con la función de raíz cuadrada de la librería Java.

2- Gestión de Referencias Bibliográficas (3/4 * 0.45, 0.45, 0.1)

(sugerencia para la realización de este ejercicio: implementar con Netbeans todo lo que el enunciado va especificando de manera concreta, dejando para una segunda lectura del mismo todo aquello que supone dejar a su discrección la solución.

Si le sobra tiempo no se vaya sin documentar debidamente el código para que Javadoc proporcione unas páginas razonables).

Las referencias bibliográficas de artículos publicados en revistas científicas contienen información como el nombre de los autores, el título, el nombre de la revista, el número o volumen, la fecha de publicación y el número de página donde comienza el artículo. En particular el estándar APA tiene este aspecto:

Banerjee, S., Danielsson, U., Dibitetto, G., Giri, S., & Schillo, M. (2018). Emergent de Sitter Cosmology from Decaying Anti-de Sitter Space. *Physical review letters*, 121(26), 261301.

a) Defina la clase **Reference** que permita almacenar los campos asociados a referencias bibliográficas:

- la LISTA de autores {"Banerjee, S.", "Danielsson, U.", "Dibitetto, G.", "Giri, S.", "Schillo, M."}
- el año de publicación. "2018"
- el título del artículo "Emergent de Sitter Cosmology from Decaying Anti-de Sitter Space"
- el nombre del medio/revista "*Physical review letters*"
- el volumen "121(26)"
- el número de la primera página del artículo. "261301"

Defina también cualquier otra clase, interfaz, enumeración, etc. que considere necesarias en función de lo que se indica a continuación para **Reference**.

- los campos serán representados mediante Strings (o una lista de Strings, en el caso de los autores).
- Contará con un único constructor vacío (sin argumentos).

- El valor de cada campo de una referencia podrá ser añadido independientemente del resto mediante él método:

```
public void add(XXX field, String value)
```

donde el tipo **XXX** depende de su diseño, mientras que, como anteriormente se ha comentado, los valores de los campos siempre serán representados mediante cadenas de caracteres.

Preste atención a los detalles, como son:

- Situaciones de excepción, por ejemplo (puede depender de su diseño):
 - recibir un valor para un campo ya cumplimentado anteriormente que no coincide con el anterior.
 - El campo indicado no existe.
 - ...
 - En caso de que se reciba un mismo autor dos o más veces no añadirlo a la lista.
 - Tener claro cual es el estado de una referencia para la no se han recibido los datos para todos los campos.
 - ...
- Deberá estar dotada de los *getters* que sean precisos.
 - Considere la herencia de Object y reescriba los métodos que considere necesarios (para los que crea que no se precisa una reescritura, añada un comentario justificándolo). En especial, reescriba el metodo **toString**, de tal manera que devuelva una cadena con el formato previamente mostrado:

Autor₁, autor₂, ..., autor_{n-1} & autor_n (año). título. revista, volumen, página.

El siguiente metodo estático debería funcionar perfectamente y mostrar la referencia:

```
public static void main(String[] args) {
    Reference r = new Reference();
    r.add(<t>,"Emergent de Sitter Cosmology from Decaying Anti-de Sitter Space");
    r.add(<a>,"Banerjee, S.");
    r.add(<a>,"Danielsson, U.");
    r.add(<a>,"Dibitetto, G.");
    r.add(<v>,"121(26)");
    r.add(<a>,"Giri, S.");
    r.add(<a>,"Schillo, M.");
    r.add(<r>,"Physical review letters");
    r.add(<f>,"2018");
    r.add(<p>,"261301");
    System.out.println(r);
}
```

Donde las especificaciones <t>, <a>, <r>, <v>, <f>, <p>, hacen referencia al modo en que su solución aporte la información sobre el campo de que se trata (título, autor, revista, volumen, fecha, y página respectivamente)

b) Defina la clase **DataBase** que represente una base de datos de referencias bibliográficas. La clase deberá cumplir con los siguientes requerimientos:

- Deberá contar con tres constructores (a partir del nombre de un fichero, un InputStream y una URL) que permitan instanciar una DataBase a partir de un texto compuesto por líneas, donde cada una contendrá el valor de un campo de una referencia, siendo el título siempre el primer campo en aparecer, como se muestra en el siguiente ejemplo (cuenta con un ejemplo con más referencias en <http://gtts.ehu.es/German/Docencia/1819/TAP/examenes/DB.txt>):

```
Title: Emergent de Sitter Cosmology from Decaying Anti-de Sitter Space
Publication: Physical review letters
Date: 2018
Page: 261301
Author: Banerjee, S.
Author: Danielsson, U.
Author: Dibitetto, G.
Volume: 121(26)
Author: Giri, S.
Author: Schillo, M.
```

```
Title: Amino acid substitution within the VP7 protein of G2 rotavirus strains
Author: M. Iturriza
Author: D. Cubitt
Date: 2001
Page: 151
Publication: Am Soc Microbiol
Author: U. Desselberger
```

Además, la base de datos deberá contar con los siguientes métodos:

- `public List<Reference> searchByTitle(String s);`
 - Devuelve una lista de referencias cuyo título contiene la cadena proporcionada (independientemente de las mayúsculas o minúsculas).
- `public List<Reference> searchByAuthor(String s);`
 - Devuelve una lista de referencias con algún autor que contenga la cadena proporcionada (independientemente de las mayúsculas o minúsculas).
- `public String bestAuthor();`
 - Devuelve el autor con mayor número de publicaciones.

c) Haciendo uso del recurso de texto <http://gtts.ehu.es/German/Docencia/1819/TAP/examenes/DB.txt> (descargándolo o accediendo a él directamente), escriba un método `main` que escriba por pantalla el número de referencias que contengan “approach” en su título, el número de referencias que contengan “new” en sus autores y el autor con mayor número de publicaciones.